

REMARKS/ARGUMENTS

Reconsideration of this application is respectfully requested.

The rejection of all pending claims 1-6, 8-9, 11-14, 16-17 and 19-21 under 35 U.S.C. §103 as allegedly being made "obvious" by Brown '551 in view of newly cited Kuruvila '315 is respectfully traversed.

As the Examiner has already acknowledged, Brown does not "explicitly teach about establishing a duplicate master, wherein the role of said duplicate master is switchable between said duplicated objects when a previous duplicate master becomes unavailable". Indeed, Brown does not even teach establishment of a duplicate master -- by any technique. To the extent that Brown teaches the need to fully recover from a system failure (column 2, lines 20-25), Brown also explicitly teaches the contemplated way to do that by creating record files on the shared server for each user -- thus permitting the historical transaction file to be re-executed during an automatic recovery of the multi-user word processing system (e.g., see column 4, lines 35-40). Attempting to maintain duplicate "masters" in the context of a multi-user dynamic real time word processing system would not appear to be a fruitful alternative. In any event, there is certainly no suggestion in either Brown or any other reference of record that would have suggested to one of ordinary skill in the art that some alternate solution (i.e., other than the one explicitly already taught by Brown) was needed or even useful in that environment.

In short, there is no motivation in Brown (or in Kuruvila for that matter) to make the modification apparently proposed by the Examiner. Furthermore, it is unclear how a selected

teaching from KuruVila in a completely non-analogous context might actually be engrafted onto the disparate Brown teaching.

Brown represents a classical server based network in which normal operation would involve single-user access only to a particular file of data. In Brown (essentially a word processing package) multi-user access is provided to a document. There is provision of a control file and user record files. User record files have version numbers and these version numbers are compared so as to facilitate the updating of the control file.

It is therefore important to realize that the topology of Brown is one in which remote terminals provide access to an all powerful server. If the server goes down, the whole network goes down. The disclosure of Brown therefore brings very little to the problem encountered by the present inventors.

In the present specification, examples are given of the type of application for which the claimed invention is relevant. These include distributed diary systems and distributed gaming systems. In these systems, most of the work is done locally. Thus, locally, a user is provided with functional programs and operational data. However, an illusion is created to the effect that all users are considering the same data. It is therefore necessary for the individual copies of data to be synchronized. In order to oversee the synchronization process, one of the user stations adopts the role of master. Other slave stations may join the network and drop out of the network at will. Similarly, it is possible for the master station to drop out of the network. When this occurs, it is necessary for the role of the master object to be switched to one of the duplicated objects. Thus, the present invention works within an environment that may be considered as

peer-to-peer with one of the peers being the master. However, each peer has the capability to become master and one of them will do so if the existing master drops out of the network. The network is therefore highly robust while maintaining synchronization.

It is important to note that applicant's operational software remains resident at the individual devices. These devices could be different therefore the nature of the software could in itself be different. Operating systems could be different for example. The duplication of data relates to data and it is this data that is synchronized and controlled under the operation of a master. The actual nature of the operations performed by the operational software does not vary; it is only the data that varies as the data is updated for example during the performance of an interactive game with distributed users.

Referring to column 1, line 19 of Kuruvila, reference is made to control software such as that used in a microwave controller, a telephone system or application software such as spreadsheets or word processors. It is also stated that the maintenance of such systems has proven to be quite costly.

In this respect, it should be noted that the present invention is not concerned with the maintenance of operational software instructions. The present invention is concerned with the maintenance and updating of operation data.

Kuruvila goes on to state that the occurrence of faults is of particular concern. Failure of server software can occur because of a problem with the hardware on which the application is being executed or because of a run-time error in the software system. It goes on to state that these software systems may have improved reliability through the provision of redundancy.

Line 20 of column 5 in Kuruvila refers to concern with providing redundancy at the software instruction level in order to obtain more robust systems. Thus, a component of the operational software is provided with sparing capabilities. This means that if a component of the operational software fails, an alternative component, identified as a spare, may be brought on line and thereby maintain operational integrity. However, it should be emphasized that this component represents program instructions and it is there to maintain the overall operation of the system. This does not relate to duplicated data objects as in the present invention.

Line 5 of column 7 of Kuruvila refers to a master being permitted to have many spares. Line 14 of column 7 also refers to a higher program version spare being used to replace the master when this is taken off the system. Again, this emphasizes the program instruction nature of the object which is distinctly different from the maintenance of user data objects.

Thus, it is not correct to say that in Kuruvila a master data object is duplicated at a plurality of slave stations. In Kuruvila, spare operational code (i.e., computer program software) is provided to replace failed code or to update code. The spare code by its very nature is different because different functionality is required whether in order to repair a failure or in order to upgrade functionality. In Kuruvila, instructions are given enhanced status to the effect that they are a spare and as such may be used to replace another component of the software. This is clearly different from the notion of duplicating user data objects at a plurality of slave stations.

Furthermore, in the present invention, the data object identified as the master is configured to maintain consistency in data between the duplicated data objects. That is to say, the duplicated object data should be the same as the data maintained by the master object.

Again, this is totally in conflict with the proposal put forward in Kuruvila. The slave programs are available to take over and must remain as originally written. You certainly would not (in Kuruvila) wish the spare programs to be maintained consistently with the current master program instructions because the very reason for it being there is to be different and self contained such that, under the appropriate circumstances, it can take over.

In the present invention, the taking over of the role as master has nothing to do with system failure or performance upgrades. A transfer of roles in which the role of master object is switched to one of the duplicated objects, only becomes necessary when the current master hardware goes off line. This is not a system failure. It may represent a user powering down their equipment or a gamer dropping out of a game. These activities do not represent system failure but are merely examples of normal system use. As such, the system must accommodate them and it achieves this by having a master object that maintains data consistency between the duplicated objects such that the master object is maintained and the copies are updated. Furthermore, the role of the master object is switched to one of the duplicated objects when a first master station becomes available.

Attention is also directed to new method claim 22 which requires maintaining a data object as a master data object at a first station, duplicating copies of the master data object at a plurality of slave stations wherein the master data object is configured to maintain consistency between data in the duplicated objects with its own data as well as maintaining such copies in an updated fashion and further requiring, when the first station becomes unavailable in normal ordinary ongoing operations, to switch the role of the master data object to one of the duplicated

DIONNE et al.
Appl. No. 09/735,925
April 25, 2006

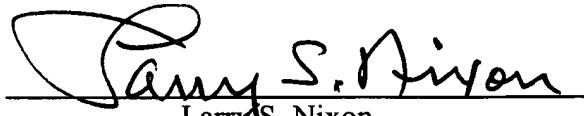
data objects. Clearly there is no possible suggestion of such features anywhere in any of the cited prior art.

Accordingly, this entire application is now believed to be in allowable condition and a formal Notice to that effect is respectfully solicited.

Respectfully submitted,

NIXON & VANDERHYE P.C.

By:



Larry S. Nixon
Reg. No. 25,640

LSN:vc
901 North Glebe Road, 11th Floor
Arlington, VA 22203-1808
Telephone: (703) 816-4000
Facsimile: (703) 816-4100